Computer Science

A GENETIC ALGORITHM FOR MULTIPROCESSOR TASK SCHEDULING WITH INTERPROCESSOR DELAY

Jessica Yantachka and Frank Moore*
Miami University Computer Science and Systems Analysis Department
230 Kreger Hall, Oxford, OH, 45056, {yantacj, moorefw}@muohio.edu

Background

Efficient multiprocessor task scheduling is a long-studied and difficult problem that continues to be a topic of considerable research. This NP-hard problem was traditionally solved using a combination of search techniques and heuristics. These solutions required a deterministic search of the solution space, which was computationally and temporally exhaustive. Genetic algorithms (GAs) are known to provide robust, stochastic solutions for numerous optimization problems. This project utilized a GA for minimizing the schedule length for a general task graph to be executed on a multiprocessor system. This implementation is scalable and adaptable to a variety of task graphs and parallel processing systems.

Implementation

The goal of this project was to extend previous research by comparing three genetic algorithms and testing their performance on complex task graphs. The project began with a careful study of three previous papers: the original GA designed by Hou, Ansari and Ren [1], an improved GA by Bohler *et al* [2], and a GA by Kramer [3] that considered realistic interprocessor delay during schedule evolution. After analyzing these algorithms, Kramer's code was used to reverse-engineer the previous algorithms. A switch was then added so the user could choose to execute any of the three algorithms from the same program file. A program was written to create arbitrarily large task graphs with a specified branching factor. Each GA was tested using graphs of 500 tasks with branching factors of 2, 4, 6, 8, and 10, and 1000 tasks with branching factors of 4, 8, 12, 16, and 20. For each task graph, tests were run for cases with 10 and 50 processors. Schedules optimized by the Hou and Bohler GAs were subsequently tested on task graphs with interprocessor delays.

Results

This research established a methodology for comparing the performance of various task scheduling algorithms. Both Bohler's algorithm and Kramer's improved upon Hou *et al*'s task scheduler by 40-55%. With a smaller number of tasks, Kramer's algorithm outperformed Bohler's by an average of 1.9%. For graphs with 1000 tasks, Bohler's algorithm actually outperformed Kramer's 40% of the time. These results indicate that consideration of interprocessor delay during schedule evolution has minimal benefit for larger and more complex task graphs. Future research will compare schedules optimized by traditional analytical techniques, such as the branch and bound algorithm, with the results obtained by these GAs.

Bibliography

[1] Hou, E., N. Ansari and H. Ren 1994. A Genetic Algorithm for Multiprocessor Scheduling, in IEEE Transactions on Parallel and Distributed Systems, Vol. 5 No. 2, pp. 113-120, Feb. 1994.

[2] Bohler, M., F. Moore, and Y. Pan 1999. Improved Multiprocessor Task Scheduling Using Genetic Algorithms, in Kumar, A. and I. Russell (eds.), Proceedings of the Twelfth International Florida AI Research Society Conference (FLAIRS-99), , pp. 140-146, AAAI Press.

[3] Kramer. G. and F. Moore 2000. A Genetic Algorithm-Based Task Scheduler for Parallel Processor Systems with Non-negligible Interprocessor Delay, in Blank, D. (ed.), Proceedings, Eleventh Midwest Artificial Intelligence and Cognitive Science Conference (MAICS-2000), pp. 57-61, AAAI Press.